

---

# Application Specific and General Software Platforms: Open Source Distribution

Bill Lorensen

GE Research

[lorensens@research.ge.com](mailto:lorensens@research.ge.com)

# Outline

---

- Background
- Open Systems
- Case Study - Insight
- Summary

# What is a Platform?

---

- Hardware
  - CPU
  - Peripherals
- Software
  - Operating System
  - Support libraries
- Services

# Visualization Platform

---

- Hardware
  - Graphics accelerator
- Software
  - Graphics APIs
    - » OpenGL
    - » Direct3D
- Services
  - Algorithms
  - Architecture / Framework

# Scanner Platform

---

- Hardware
  - CT/MRI scanner
- Services
  - Scan
  - Recon
  - Archive
  - Display
  - Film

---

# Moving from Closed Software to Open Software

# From [www.opensource.org](http://www.opensource.org)

---

“When programmers can read, redistribute, and modify the source code for a piece of software, the software evolves. People improve it, people adapt it, people fix bugs. And this can happen at a speed that, if one is used to the slow pace of conventional software development, seems astonishing.”

# Why Open?

---

- Protect software investment of funding organizations
- High quality software is possible
- Objective peer review
- Software outlives the developer
- Higher interoperability
- Access to a pool of developers that know the code
- Mix between custom and COTS code



# Open Software

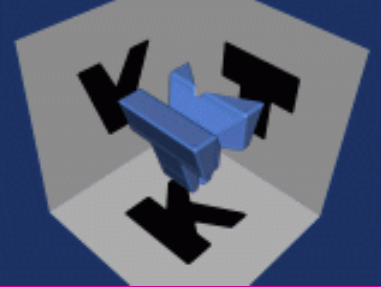
---

- Open API's
  - OpenGL
  - DICOM
  - Java
- Open Source
  - Emacs
  - Apache
  - Linux
  - OpenDX
  - vtk – Visualization Toolkit
  - itk – Insight Toolkit
  - vxl – Computer Vision Library
  - Cgal – Computational geometry
  - Slicer – surgical planning

# Open Software Business Models

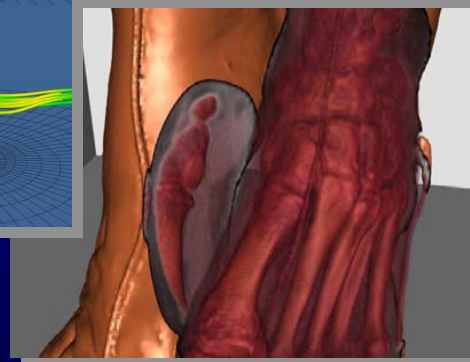
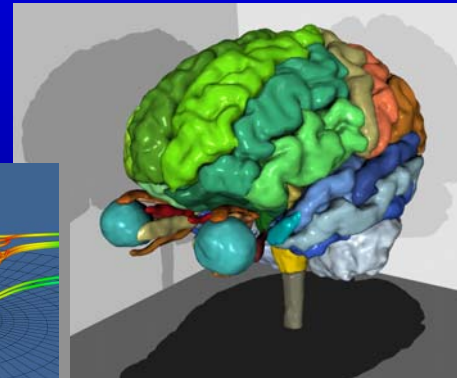
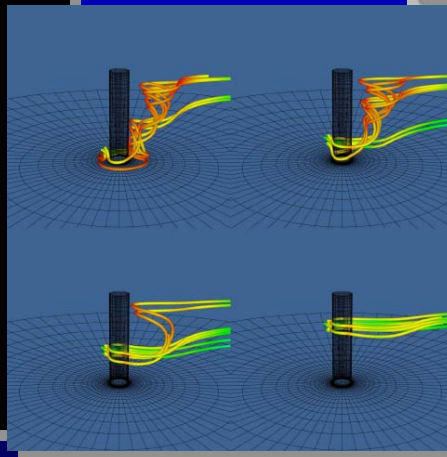
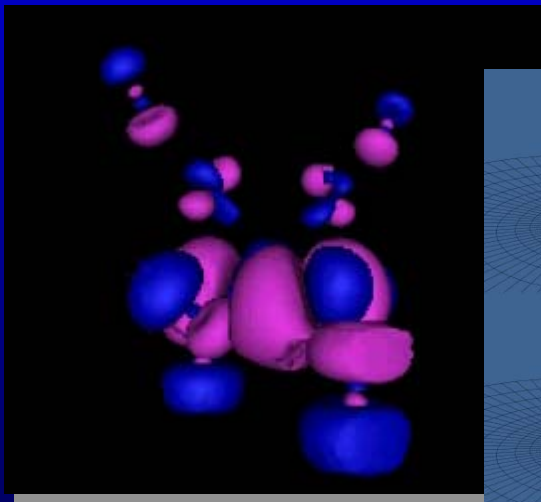
---

- Packaging, Service, Integration
  - Linux - Redhat
  - vtk - Kitware
- Hardware acceleration
  - VolumePro from TeraRecon
  - Mercury Computer's accelerated vtk
- Open plus Proprietary
  - GE Direct3D



# Visualization Toolkit - vtk

- Open source toolkit for scientific visualization, computer graphics, and image processing



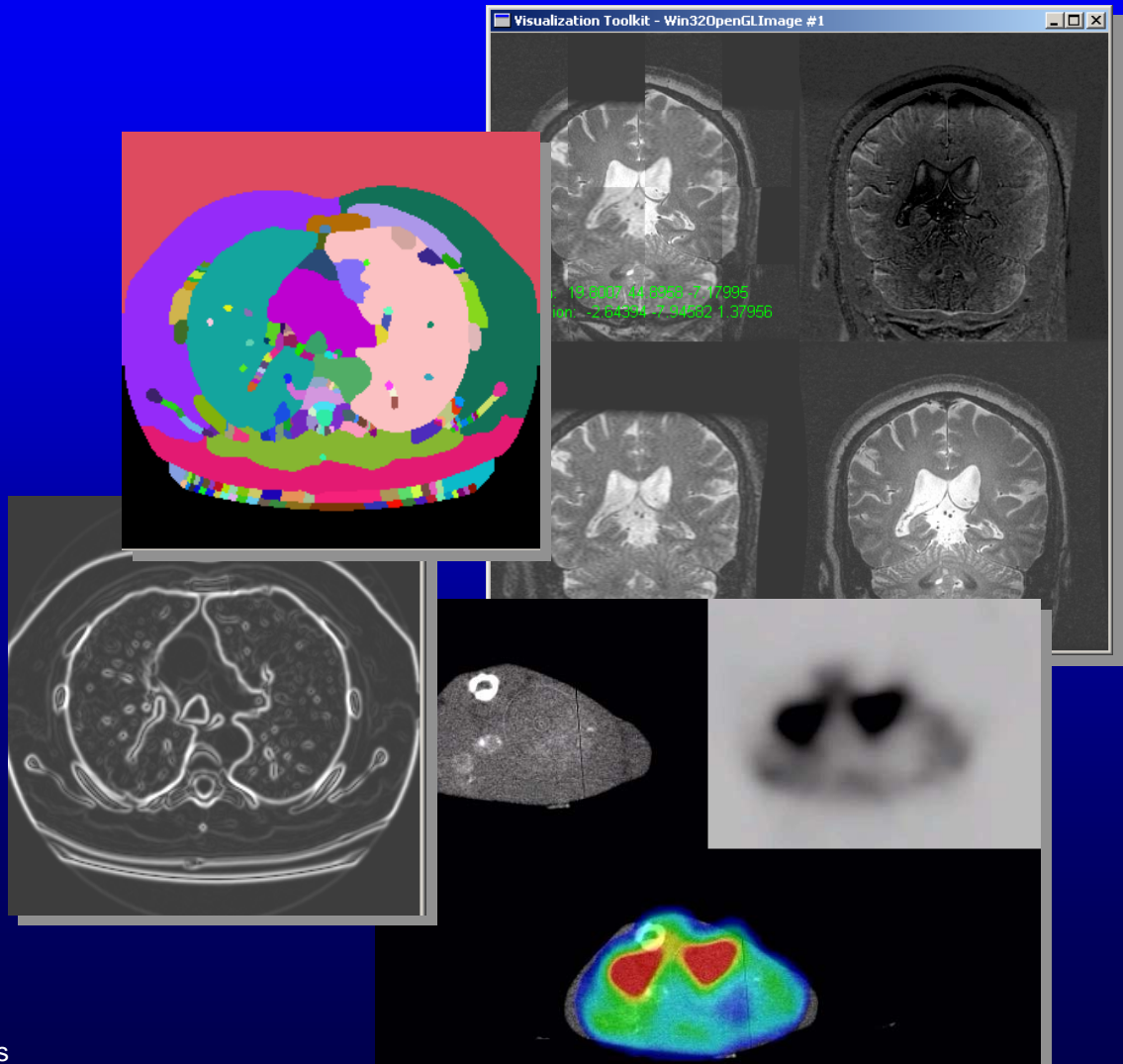
[www.vtk.org](http://www.vtk.org)



# Insight Toolkit - itk

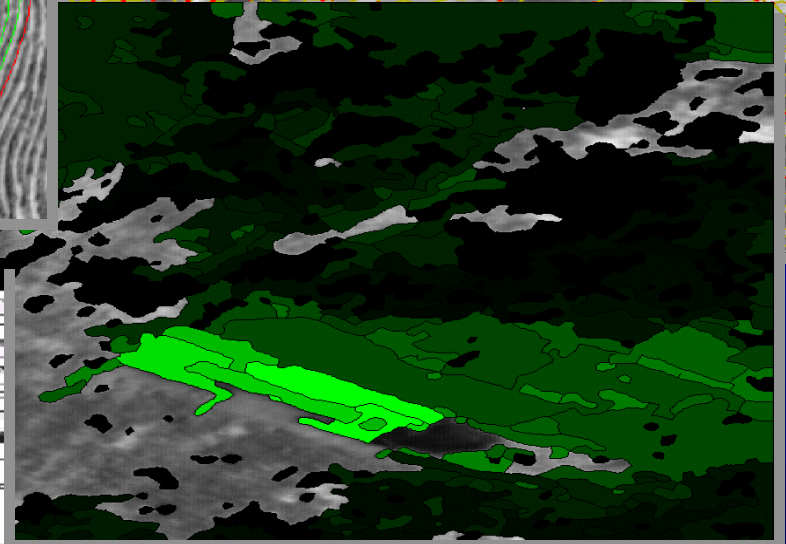
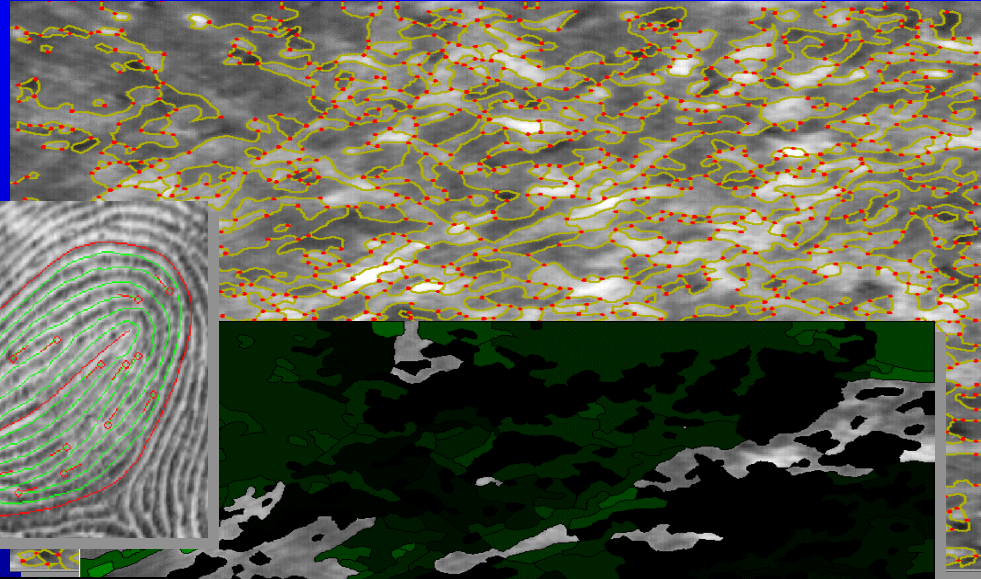
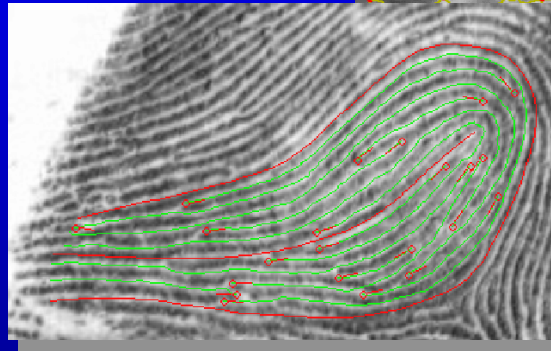
- Seeded by National Library of Medicine funding
- N-dimensional image processing
- Segmentation
- Registration

[www.itk.org](http://www.itk.org)



# Computer Vision Library - vxI

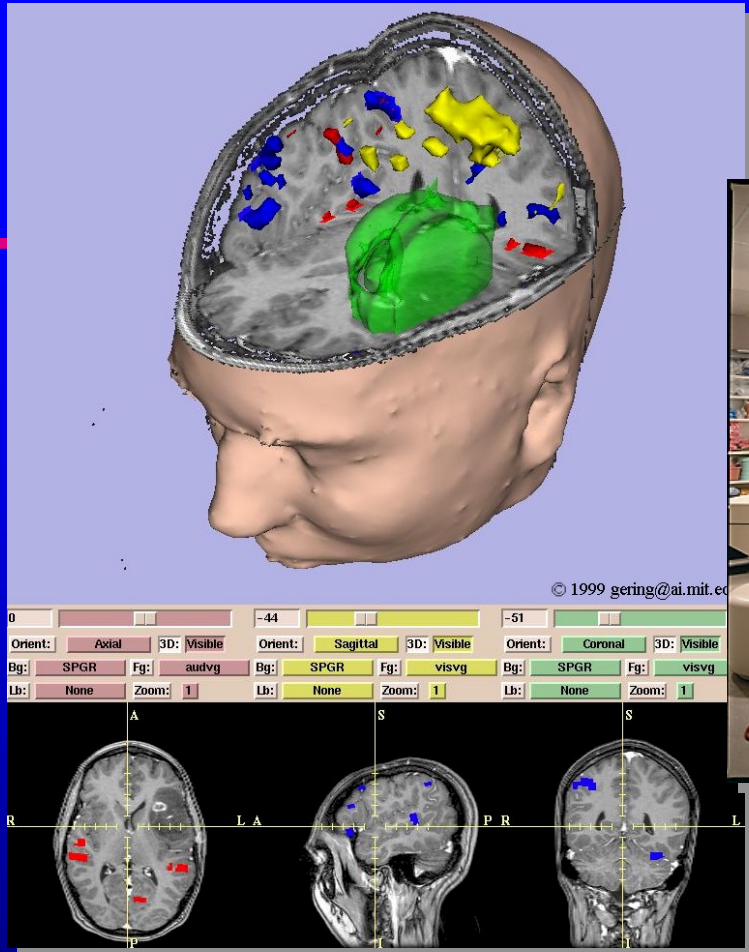
- Based on Target jr and Image Understanding Environment
- Numerics
- Imaging
- Geometry
- Camera models



[vxl.sourceforge.net](http://vxl.sourceforge.net)



# Slicer



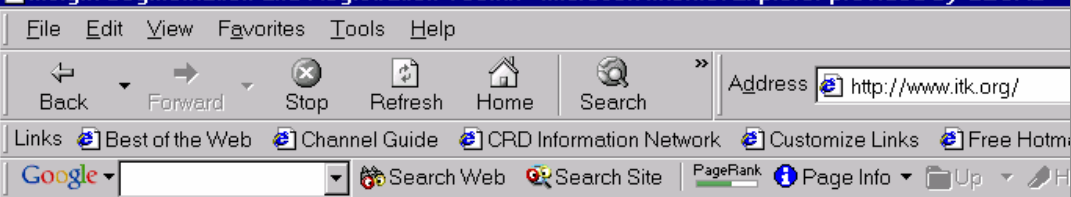
- Developed by Brigham and Womens Surgical Planning Lab
- User interface plus plug-ins for applications
  - Segmentation
  - Registration
  - Image Guidance

[www.slicer.org](http://www.slicer.org)

---

# An Open Source Case Study

## Insight



# NLM Insight

## Segmentation & Registration Toolkit

---

[Home](#)  
[About](#)  
[Sponsors](#)  
[Download](#)  
[Documentation](#)  
[FAQ](#)  
[Mailing Lists](#)  
[Testing](#)  
[News](#)  
[Bug Tracking](#)  
[Data](#)  
[Related Software](#)  
[Examples](#)  
[Copyright](#)  
[Developers Only](#)

## Welcome

to the National Library of Medicine Insight Segmentation and Registration Toolkit (ITK). ITK is an open-source software system to support the [Visible Human Project](#). Currently under active development, ITK employs leading-edge segmentation and registration algorithms in two, three, and more dimensions.

The Insight Toolkit was developed by six principle organizations, three academic ([UNC Chapel Hill](#), [University of Utah](#), [University of Pennsylvania](#)) and three commercial ([GE Corporate R&D](#), [Kitware](#), and [Insightful](#)). Additional team members include Harvard Brigham & Women's Hospital, University of Pittsburgh, and Columbia University. The funding for the project is from the National Library of Medicine at the National Institutes of Health. NLM in turn was supported by member institutions of NIH (see [sponsors](#)).

The goals for the project include the following:

- Support the Visible Human Project.
- Establish a foundation for future research.
- Create a repository of fundamental algorithms.
- Develop a platform for advanced product development.
- Create conventions for future work.
- Grow a self-sustaining community of software users and developers.



THE VISIBLE HUMAN PROJECT®

# Insight Segmentation and Registration Toolkit



# What is it?

---

- A common Application Programmers Interface (API).
  - A framework for software development
  - A toolkit for registration and segmentation
  - An Open Source resource for future research
- A validation model for segmentation and registration.
  - A framework for validation development
  - Assistance for algorithm designers
  - A seed repository for validated segmentations

# Insight - Recipe for Success

---

- Vision
- Strong Core Team
- Openness
- Core Architecture
- Light Weight Software Engineering
- Community Support
- Funding

# The NLM Vision

---

Create a *dynamic, self-sustaining, public domain* and *extensible* toolkit that will empower researchers throughout the world to develop new segmentation and registration algorithms and create new applications that leverage the NLM's investment in the Visible Human Male and Female data sets

# The Insight Team



- GE Research/Brigham and Womens
  - Architecture, algorithms, testing, validation
- Kitware
  - Architecture, user community support
- Insightful/UPenn
  - Statistical segmentation, mutual information registration, deformable registration, level sets
  - Beta test management
- Utah
  - Level sets, low level image processing
- UNC/Pitt
  - Image processing, registration, high-dimensional segmentation
- UPenn/Columbia
  - Deformable surfaces, fuzzy connectedness, hybrid methods

# Openness

---

- From the start, NLM recognized the value of Open Source software
- There are NO restrictions on the software
- The Team has embraced openness

---

# In search of a new software development process

# Extreme Programming

Cover Feature

## Embracing Change with Extreme Programming



**Extreme Programming turns the conventional software process sideways. Rather than planning, analyzing, and designing for the far-flung future, XP programmers do all of these activities—a little at a time—throughout development.**

*Kent Beck*  
First Class  
Software

In the beginning was the waterfall (Figure 1a). We would get the users to tell us once and for all exactly what they wanted. We would design the system that would deliver those features. We would code it. We would test to make sure the features were delivered. All would be well.

shows, the waterfall begat iterations.

The waterfall model didn't just appear. It was a rational reaction to the shocking measurement that the cost of changing a piece of software rose dramatically over time. If that's true, then you want to make the biggest, most far-reaching decisions as early in the

# A Light Weight Software Engineering Process

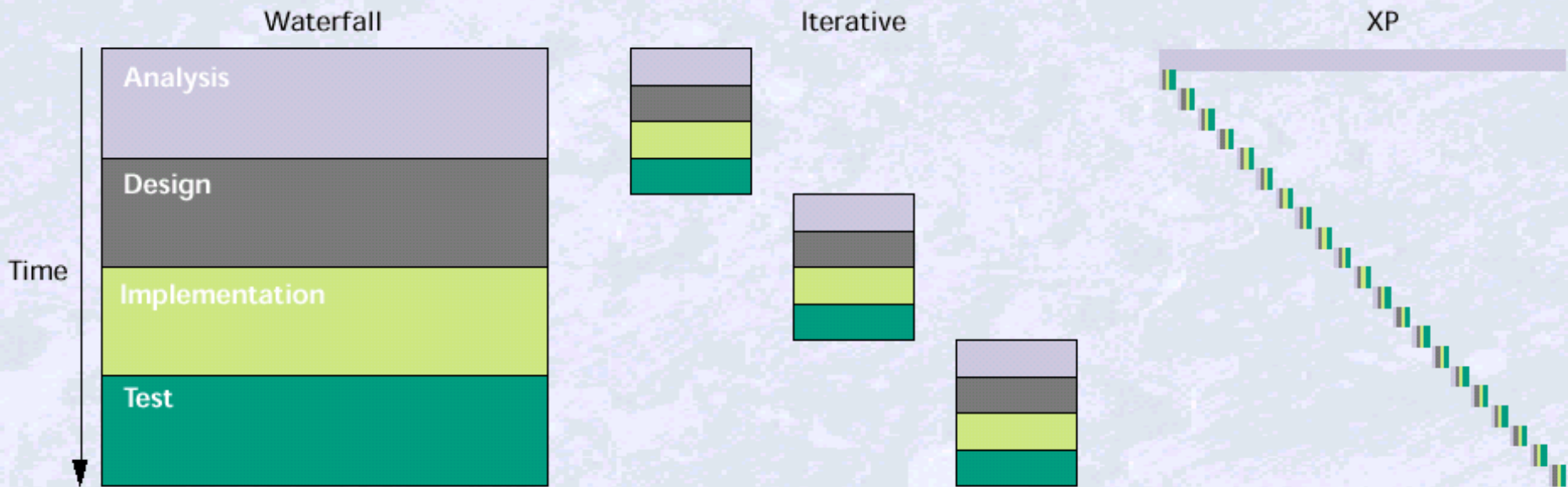
---

- Based on the new Extreme Programming process
  - High intensity design, test, implement cycle
  - Supported with web-enabled tools
  - Automated testing integrated with the software development



# Extreme Programming

Compresses the standard analyze, design, implement, test cycle into a continuous process



# A process supported by a suite of portable, open source tools

---

- Apache, perl, php
  - Web services
- CVS
  - Revision control
- Doxygen
  - Automated documentation
- CMake
  - Cross-platform program build
- Dart
  - Continuous and distributed test reporting

# Insight - Open Source Products

**NLM Insight**  
Segmentation & Registration Toolkit

**Welcome** to the National Library of Medicine Insight Segmentation and Registration Toolkit (ITK). ITK is an open-source software system to support the [Visible Human Project](#). Currently under active development, ITK employs leading-edge segmentation and registration algorithms in two, three, and more dimensions.

The Insight Toolkit was developed by six principle organizations, three academic (UNC Chapel Hill, University of Utah, University of Pennsylvania) and three commercial (GE Corporate R&D, Kitware, and Insight). Additional team members include Harvard Brigham & Women's Hospital, University of Pittsburgh, and Columbia University. The funding for the project is from the National Library of Medicine at the National Institutes of Health. NLM in turn was supported by member institutions of NIH (see [sponsors](#)).

The goals for the project include the following:

- Support the Visible Human Project
- Establish a foundation for open-source software development
- Create a repository of software and documentation
- Develop a platform for software development
- Create conventions for software development
- Grow a self-sustaining community of developers.

**CMake**  
Cross-platform Make

**Welcome** to CMake, the cross-platform, open-source make system. CMake is used to control the software compilation process using simple platform and compiler independent configuration files. CMake generates native makefiles and

**CABLE**

**Welcome** to the CABLE home page.

**Introduction**

Many software professionals recognize the value of hybrid compiled/interpreted software environments. Compiled languages such as C and C++ provide efficiency, speed, and flexible data structure representation. Interpreted languages such as Tcl or Python remove the compile/link cycle from the development process, are indispensable for prototyping, and provide a multitude of packages for creating GUI's, numerics, and web access.

The creation of such hybrid environments is difficult. Typically C/C++ developers will manually add the interpreted language binding, or use semi-automatic tools such as SWIG. In some cases, custom wrapper generators have been created that are specialized to a particular system, such as the Visualization Toolkit (VTK). However, no fully automatic, general wrapping system has been available. As a result, mixed language systems are less common than might be expected, requiring excessive resources to develop.

Kitware, Inc. has developed an open-source wrapping package called CABLE (CABLE Automates Bindings for Language Extension). It is a tool designed to automatically generate bindings to C++ classes for use in interpreted languages. This system works in conjunction with GCC-XML, an extension to the GCC compiler. GCC-XML, also developed by Brad King at Kitware, is used to parse arbitrarily complex C++ code and then produce an XML representation. This representation is then processed by CABLE to generate wrappers. See the page on [Running CABLE](#) for more details.

CABLE was developed by Brad King at [Kitware](#) as part of the [NLM Insight Segmentation and Registration Toolkit](#) project.

**Dart**  
Tests, Reports, and Dashboards

**Welcome** to Dart, an open-source, distributed, software quality system. Dart allows software projects to be tested at multiple sites in multiple configurations (hardware, operating systems, compilers, etc.). Results from a build/test sequence are transmitted to a central server using standard internet protocols. The server produces concise dashboards, summarizing the current state of a software system. The dashboards link to detailed reports on inter- and intra-configuration results. Testing results are tracked over time, allowing developers to trace the history of development.

Dart empowers every developer in a distributed software development team to track the quality of their project. Furthermore, Dart allows a developer to experiment with a locally modified version of their software and submit the results of their experiments to a central dashboard for all developers to see.

Dart consists of a server and several client machines. Dart clients build and test a software project and submit build logs and test results to the Dart server. Dart clients encode build logs and test results in XML and transmit these reports to the Dart server over the internet. The Dart server summarizes the information from the clients and produces dashboards and reports.

The Dart dashboard (or dartboard) mechanism runs on a 24 hour clock, where each day a new dashboard is opened and closed. While a dashboard is open, Dart accepts submission of nightly and experimental builds. Nightly builds are timestamped to a source code repository at 3am EST. Since the nightly builds are locked to a timestamp, they can actually be performed at any time during which that day's dashboard is open for submissions. Experimental builds represent any other build/test sequence that is not locked to the nightly build cycle. These may include continuous builds that are triggered whenever something in the repository changes or may be experimental builds reporting on a user's locally modified version of the software. Please go [here](#) to learn more about Dart.

Dart was developed by [GE Corporate Research and Development](#) as part of the [NLM Insight Segmentation and Registration Toolkit](#) project. Other sponsors include the [Insight](#), [VTK](#), and [VXL](#) open source software communities.

The goals for Dart include the following:

- Develop an open source, cross-platform tool to perform automated builds and tests
- Distribute testing across multiple computers at multiple sites
- Produce summary dashboards
- Correlate results from different configurations or across time
- Allow experimental builds and continuous builds to be summarized with "nightly" testing
- Allow anyone to test a software project
- Allow anyone to view the current state of the system

# Insight - Development Cycles

---

- Daily – dashboard
- Weekly – telephone conferences
- Bi-weekly – architecture reviews
- Quarterly – developer meetings
- Yearly – work assignments

# Extreme Programming

## Daily Testing Is The Key

---

- Testing anchors the development process (Dart)
- Developers monitor the testing dashboard constantly
- Problems are identified and fixed immediately
- Developers receive e-mail if they “Break the Build”



2 Files Changed by 2 Authors as of 2002-02-11 1:00:00 EST

No Doxygen information available!

## Nightly Builds

Site	Build Name	Update	Build		Test				Build Date	Submission Date
			Errors	Warnings	NotRun	Failed	Passed	NA		
caleb.crd	SunOS-5.6-c++		0	151	0	5	219	0	Mon Feb 11 02:32:29 EST 2002	Mon Feb 11 05:07:47 EST 2002
ct02_oc.crd	IRIX64-6.5-CC-64		0	19	0	4	220	0	Mon Feb 11 02:30:30 EST 2002	Mon Feb 11 04:48:47 EST 2002
esopus.crd	SunOS-5.7-c++		0	190	0	4	220	0	Mon Feb 11 02:24:27 EST 2002	Mon Feb 11 08:58:45 EST 2002
kitware.com	Linux-gcc3.0-beta0_5		0	167	0	1	205	18	Mon Feb 11 01:36:12 EST 2002	Mon Feb 11 04:56:44 EST 2002
kulu.crd	IRIX64-6.5-CC-n32		0	19	0	3	221	0	Mon Feb 11 02:31:19 EST 2002	Mon Feb 11 06:30:57 EST 2002
moxel2.crd	WinNT-IntelC++50		50	161	0	3	221	0	Mon Feb 11 01:40:16 EST 2002	Mon Feb 11 08:28:21 EST 2002
moxel3.dyn.crd	Linux-2.4.0-0.99.11smp-c++		0	119				224	Mon Feb 11 02:29:39 EST 2002	Mon Feb 11 04:13:13 EST 2002
pragmatic.crd	Linux-2.4.2-2smp-c++	1	0	35	0	4	220	0	Mon Feb 11 02:24:29 EST 2002	Mon Feb 11 03:57:53 EST 2002
sextant.crd	CYGWIN_NT-4.0-1.3.6_0.47_3_2_-c++		168	0	4	29	191	0	Mon Feb 11 02:04:04 EDT 2002	Mon Feb 11 05:29:57 EST 2002

## Continuous Builds

Site	Build Name	Update	Build		Test				Build Date	Submission Date
			Errors	Warnings	NotRun	Failed	Passed	NA		
sextant.crd	WinNT-VC++60-Continuous	5	0	3	0	3	221	0	Mon Feb 11 13:32:02 EST 2002	Mon Feb 11 14:19:05 EST 2002
caleb.crd	SunOS-5.7-c++-Continuous		0	124	0	5	219	0	Mon Feb 11 11:35:04 EST 2002	Mon Feb 11 14:02:11 EST 2002
sextant.crd	WinNT-VC++60-Continuous	9	0	1	0	3	221	0	Mon Feb 11 11:35:25 EST 2002	Mon Feb 11 12:00:13 EST 2002
pragmatic.crd	Linux-2.2.14-5.0smp-c++-Continuous	12	0	62	0	4	220	0	Mon Feb 11 11:37:34 EST 2002	Mon Feb 11 13:55:51 EST 2002
sextant.crd	WinNT-VC++60-Continuous	2	0	0	0	3	221	0	Mon Feb 11 10:00:57 EST 2002	Mon Feb 11 10:19:58 EST 2002
sextant.crd	WinNT-VC++60-Continuous	1	0	0	0	3	221	0	Mon Feb 11 09:31:16 EST 2002	Mon Feb 11 09:53:44 EST 2002
caleb.crd	SunOS-5.7-c++-Continuous		0	151	0	5	219	0	Mon Feb 11 08:50:37 EST 2002	Mon Feb 11 11:17:28 EST 2002
pragmatic.crd	Linux-2.2.14-5.0smp-c++-Continuous	12	0	76	0	4	220	0	Mon Feb 11 08:53:39 EST 2002	Mon Feb 11 11:15:15 EST 2002
sextant.crd	WinNT-VC++60-Continuous	8	0	0	0	3	221	0	Mon Feb 11 08:31:35 EST 2002	Mon Feb 11 08:54:45 EST 2002
sextant.crd	WinNT-VC++60-Continuous	2	0	0	0	3	221	0	Mon Feb 11 07:59:22 EST 2002	Mon Feb 11 08:18:18 EST 2002
sextant.crd	WinNT-VC++60-Continuous	2	0	0	0	3	221	0	Mon Feb 11 07:28:17 EST 2002	Mon Feb 11 07:52:06 EST 2002
caleb.crd	SunOS-5.7-c++-Continuous		0	151	0	5	219	0	Mon Feb 11 06:10:56 EST 2002	Mon Feb 11 08:36:07 EST 2002
pragmatic.crd	Linux-2.2.14-5.0smp-c++-Continuous	1	0	81	0	4	220	0	Mon Feb 11 06:06:54 EST 2002	Mon Feb 11 08:10:08 EST 2002

## Experimental Builds

Site	Build Name	Update	Build		Test				Build Date	Submission Date
			Errors	Warnings	NotRun	Failed	Passed	NA		
pragmatic.crd	Linux-2.4.2-2smp-c++		0	0	0	8	216	0	Mon Feb 11 12:46:16 EST 2002	Mon Feb 11 13:02:26 EST 2002



# Dashboard - Mon Feb 11 16:00:34 EST 2002

Monday, February 11 2002

[Dashboard](#) [Updates](#) [Tests](#) [Today](#) [CVS](#) [Doxygen](#) [Bugs](#) [Home](#)

[2 Files Changed](#) by 2 Authors as of 2002-02-11 1:00:00 EST

No Doxygen information available!

## Nightly Builds

Site	Build Name	Update	Build		Test		
			Errors	Warnings	NotRun	Failed	Passed
caleb.crd	SunOS-5.6-c++		0	<a href="#">151</a>	0	<a href="#">5</a>	<a href="#">219</a>
ct02_oc.crd	IRIX64-6.5-CC-64		0	<a href="#">19</a>	0	<a href="#">4</a>	<a href="#">220</a>
esopus.crd	SunOS-5.7-c++		0	<a href="#">190</a>	0	<a href="#">4</a>	<a href="#">220</a>
kitware.com	Linux-gcc3.0-beta0_5		0	<a href="#">167</a>	0	<a href="#">1</a>	<a href="#">205</a>
kulu.crd	IRIX64-6.5-CC-n32		0	<a href="#">19</a>	0	<a href="#">3</a>	<a href="#">221</a>
moxel2.crd	WinNT-IntelC++50		50	<a href="#">161</a>	0	<a href="#">3</a>	<a href="#">221</a>
moxel3.dyn.crd	Linux-2.4.0-0.99.11smp-c++		0	<a href="#">119</a>			
pragmatic.crd	Linux-2.4.2-2smp-c++	<a href="#">1</a>	0	<a href="#">35</a>	0	<a href="#">4</a>	<a href="#">220</a>
sextant.crd	CYGWIN_NT-4.0-1.3.6_0.47_3_2_-c++		<a href="#">168</a>	0	<a href="#">4</a>	<a href="#">29</a>	<a href="#">191</a>



# Someone broke the build!

**Broken continuous build for ITK pragmatic.crd/Linux-2.4.2-2smp-c++-Continuous - Message (Pl...**

File Edit View Insert Format Tools Actions Help

Reply Reply to All Forward

From: Lorensen, William E (Research) Sent: Wed 9/11/2002 8:00 AM  
 To: Blezek, Daniel J (Research); avants@gradient.cis.upenn.edu  
 Cc: Lorensen, William E (Research); Miller, James V (Research)  
 Subject: Broken continuous build for ITK pragmatic.crd/Linux-2.4.2-2smp-c++-Continuous

A continuous build has been broken for ITK and you have been identified as one of the authors who have checked in changes that are part of this build.

Errors for this build are here:  
<http://www.itk.org/Testing/Sites/pragmatic.crd/Linux-2.4.2-2smp-c++-Continuous/20020911-1005-Continuous/BuildError.html>

The changes for this build are here:  
<http://www.itk.org/Testing/Sites/pragmatic.crd/Linux-2.4.2-2smp-c++-Continuous/20020911-1005-Continuous/Update.html>

The dashboard for the day is here:  
<http://www.itk.org/Testing/Dashboard/20020911-0500-Nightly/Dashboard.html>

If you have any questions about this email, please contact the ITK Continuous build monitors: lorensen@crd.ge.com millerjv@crd.ge.com

## Continuous Builds

Site	Build Name	Update	Error	Build Status	Build Time	Build Time
caleb.crd	SunOS-5.7-c++-Continuous		0	12	10:17:13 EDT 2002	10:41:25 EDT 2002
ct02_oc.crd	IRIX64-6.5-CC-n32-Continuous		0	5	Wed Sep 11 10:17:21 EDT 2002	Wed Sep 11 10:45:04 EDT 2002
pragmatic.-crd	Linux-2.4.2-2smp-c++-Continuous	2	0	3	Wed Sep 11 10:11:15 EDT 2002	Wed Sep 11 10:36:47 EDT 2002
caleb.crd	SunOS-5.7-c++-Continuous		1	13	Wed Sep 11 06:13:32 EDT 2002	Wed Sep 11 08:26:44 EDT 2002
ct02_oc.crd	IRIX64-6.5-CC-n32-Continuous		1	13	Wed Sep 11 06:13:09 EDT 2002	Wed Sep 11 07:03:31 EDT 2002
pragmatic.-crd	Linux-2.4.2-2smp-c++-Continuous	4	2	3	Wed Sep 11 06:07:59 EDT 2002	Wed Sep 11 07:45:46 EDT 2002



## Color legend

Passed Failed Not Run Not Applicable

Test ( sort by )

Status ▼

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	NFP	NA			
./Testing/Code/BasicFilters/itkBloxBoundaryPointImageTest	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	6	0	17	0	
./Testing/Code/BasicFilters/itkBloxCoreAtomTest	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	6	0	17	0	
./Testing/Code/Common/itkCommonHeaderTest	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	6	0	17	0	
./Testing/Code/Common/itkRT3DTransformTest	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	3	0	20	0	
./Testing/Code/Algorithms/itkMultiResolutionPDEDeformableRegistrationTest	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	1	5	17	0	
./Testing/Code/Algorithms/itkVoronoiSegmentationImageFilterTest	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	1	0	22	0	
./Testing/Code/Algorithms/itkMultiResolutionPyramidImageFilterTest	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	0	7	16	0	
./Testing/Code/Algorithms/itkCurvatureFlowTest	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	0	5	18	0	
./Testing/Code/Algorithms/itkDemonsRegistrationFilterTest	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	0	5	18	0	
./Testing/Code/Algorithms/itkMultiResolutionMutualInformationAffineRegistrationTest	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	0	5	18	0	
./Testing/Code/Algorithms/itkMultiResolutionMutualInformationRigidRegistrationTest	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	0	5	18	0	
./Testing/Code/Algorithms/itkVectorFuzzyConnectednessImageFilterTest	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	0	5	18	0	
./Testing/Code/BasicFilters/itkEigenAnalysis2DImageFilterTest	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	0	5	18	0	
./Testing/Code/Algorithms/itkLevelSet2DEquationTest	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	0	4	19	0	
./Testing/Code/Algorithms/itkLevelSetShapeDetectionTest	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	0	4	19	0	
./Testing/Code/Algorithms/itkRecursiveMultiResolutionPyramidImageFilterTest	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	0	4	19	0	
./Testing/Code/BasicFilters/itkWrapPadImageTest	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	0	4	19	0	
./Testing/Code/BasicFilters/itkMirrorPadImageTest	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	0	2	21	0	
./Testing/Code/BasicFilters/itkDerivativeImageFilterTest	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	0	1	22	0	
./Testing/Code/BasicFilters/itkStreamingImageFilterTest2	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	0	1	22	0	
./Testing/Code/Common/itkSplineKernelTransformTest	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	0	1	22	0	
./Testing/Code/Numerics/vxl/vnl/tests/test_fft1d	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	0	1	22	0	
./Testing/Code/Numerics/vxl/vnl/tests/test_fft2d	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	0	1	22	0	
./Testing/Code/Algorithms/itkCurvatureFlowTestTcl.tcl	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	0	0	1	22	
./Testing/Code/Common/itkRGBPixelTest	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	0	0	1	22	
./Testing/Code/Algorithms/itkAlgorithmsHeaderTest	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	0	0	23	0	
./Testing/Code/Algorithms/itkDeformableTest	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	0	0	23	0	
./Testing/Code/Algorithms/itkExtractMeshConnectedRegionsTest	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	0	0	23	0	
./Testing/Code/Algorithms/itkFastMarchingTest	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	0	0	23	0	
./Testing/Code/Algorithms/itkFuzzyConnectednessImageFilterTest	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	0	0	23	0	
./Testing/Code/Algorithms/itkGeodesicActiveContoursTest	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	0	0	23	0	
./Testing/Code/Algorithms/itkGibbsTest	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	0	0	23	0	
./Testing/Code/Algorithms/itkImageMomentsTest	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	0	0	23	0	
./Testing/Code/Algorithms/itkImageToImageAffineMeanSquaresGradientDescentRegistrationTest	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	0	0	23	0	
./Testing/Code/Algorithms/itkImageToImageAffineMeanSquaresRegularStepGradientDescentRegistrationTest	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	0	0	23	0	
./Testing/Code/Algorithms/itkImageToImageAffineMutualInformationGradientDescentRegistrationTest	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	0	0	23	0	
./Testing/Code/Algorithms/itkImageToImageAffineNormalizedCorrelationGradientDescentRegistrationTest	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	0	0	23	0	
./Testing/Code/Algorithms/itkImageToImageAffineNormalizedCorrelationRegularStepGradientDescentRegistrationTest	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	0	0	23	0	
./Testing/Code/Algorithms/itkImageToImageAffinePatternIntensityRegularStepGradientDescentRegistrationTest	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	0	0	23	0	
./Testing/Code/Algorithms/itkImageToImageRigidMutualInformationGradientDescentRegistrationTest	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	0	0	23	0	
./Testing/Code/Algorithms/itkImageToImageTranslationMeanSquaresGradientDescentRegistrationTest	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	0	0	23	0	
./Testing/Code/Algorithms/itkImageToImageTranslationMeanSquaresRegularStepGradientDescentRegistrationTest	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	0	0	23	0	
./Testing/Code/Algorithms/itkImageToImageTranslationNormalizedCorrelationGradientDescentRegistrationTest	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	0	0	23	0	

# Insight - Building a Community

---

- Initial community of consortium members
- Outreach to other groups
  - Siggraph
  - Digital Human
  - DOE Genomes to Life
  - IEEE Visualization
  - NSF Shape Modeling Workshop
  - Supercomputer Conference
- Public Beta, February 2002
- Release 1.0, October 2002
- RFP for algorithms and data, July 2002

# Insight - Funding

---

- Seed funding by NLM and its co-sponsors
  - Establish an architecture
  - Implement a representative set of algorithms
  - Produce frameworks to accommodate new algorithms
  - Define a development process
  - Establish a community support mechanism
- Continued support will be needed
  - Outreach to other communities
  - Create applications
  - More algorithms to fill gaps
  - Infrastructure support

# Insight - Observations

---

- Good mix of commercial and academic
- Communication is critical
- The daily rhythm of Extreme Programming
- The Whole >>> Sum of the parts
- This process can (and should) be repeated

# Menu for Success

---

- A Community with a common vision
- A pool of talented and motivated developers
- A mix of academic and commercial
- An organized, light weight approach to software development
- A leadership structure
- Communication
- A business model

# Resources

---

- [www.opensource.org](http://www.opensource.org)
- [sourceforge.net](http://sourceforge.net)
- [www.itk.org](http://www.itk.org)
- [www.vtk.org](http://www.vtk.org)
- [www.slicer.org](http://www.slicer.org)



---

# Application Specific and General Software Platforms: Open Source Distribution

Bill Lorensen

GE Research

[lorensens@research.ge.com](mailto:lorensens@research.ge.com)